

(یہ) کیا (عقل کی بات ہے کہ) تم لوگوں کو نیکی کرنے کو کہتے ہو اور اپنے تئیں فراموش کئے دیتے ہو، حالانکہ تم کتاب (خدا) بھی پڑھتے ہو۔ کیا تم سمجھتے نہیں؟ - البقرة آیت 44

Age is an issue of mind over matter. If you don't mind, it doesn't matter. Mark Twain

Lecture 14

Methods/ Function Creation with Examples

In this lecture we will extend our discussion on creation of user defined function/ methods and see many examples. In the last we will see an example with and without function.

Method / Function Creation

Once again talking about why there is need to create more function/ methods? Simply speaking whenever you are repeating a code you should think of avoiding repetition and defining a new function a solution available. Secondly dividing code into functions is a natural way to distribute tasks when distributing a big task. Every different task can be a written as separate function even if it is not repeated yet it may be in future. Any code you write in **main** function can be written in any other function. For the matter of fact **main** is also a function and only difference from other function is every program starts from **main** function and other functions can be called from **main** function. Normally functions don't print something rather they return value(s). Consider following functions:

```
public static void cube(int n){
    System.out.println(n*n*n);
}
```

```
public static void avg(int a, int b){
    System.out.println((a+b)/2.0);
}
```

Both functions are useful if we want to see **cube** of any number or **average** of any two numbers but of no use if we want to use their results for further manipulation. Like if we have to find value of y which is equal to $\frac{2x^3}{5}$, we can't use cube function because cube function only shows value and value calculated by function is not available outside the function, therefore, it is required that function should return value and printing is left on the caller, if required. Therefore we are rewriting same functions with return statement instead of print:

```
public static int cube(int n){
    return n*n*n;
}
```

```
public static double avg(int a, int b){
    return (a+b)/2.0;
}
```

Other reasons for not printing inside function and deferring printing is that user may not want to print intermediate results but final results like there may be a function to calculate tax but user may want to show net salary only. Secondly it is also possible that user want to show result later on and in some specific format. Therefore, remember if a function is performing some computation it will return results without printing and function will print only if the function has task of printing. Like there may be separate functions to print date in different formats:

```
... void printDate1(int d, int m, int y){
    ? (d+":"+m+": "+y);
}
```

```
... void printDate2(int d, int m, int y){
    ? (d+"/"+m+"/"+y);
}
```

Now we will discuss some functions having computations. We are discussing a function to count spaces in String. Here String is input of the function; whereas; count (int) is the output of the function:

```
public static int countSpaces(String s){
    int count=0,i;
    for (i=0;i<s.length();i++)
        if (s.charAt(i)==' ')
            count++;
    return count;
}
```

Another function is to find number of digits in any number. The function is so simple that you can't think of:

```
public static int countDigits(int n){
    String s="" + n;
    return s.length();
}
```

A comparatively complex function is to convert decimal number into binary, where function has number as input in decimal form and output is also number in binary form:

```
public static int toBinary(int dec){
    int bin;
    String s="";
    while (dec!=0){
        s=dec%2+s;
        dec=dec/2;
    }
    bin=Integer.parseInt(s);
    System.out.println("Binary Number is:"+bin);
}
```

A very simple but might be confusing function is to write round function. Where function input is double value and output is integer value. The idea is to get integer value of give double value and find the difference of double value and corresponding integer value. If difference is less than 0.5 return integer value, otherwise return 1+integer value. See the code:

```
public static int round(double dVal){
    int iVal=(int)dVal;
    if (dVal-iVal<0.5)    return iVal;
    else                  return iVal+1;
}
```

Now let's start a small application with and without extra function. Here we are going to simulate 5 overs cricket match where each team has 6 players. For the sake of simplicity we are assuming that both teams will play maximum 5 overs or less in case all team out before 5 overs. At the end we will compare scores of both teams and declare winner or in case of equal score match is draw. Here we are using random number to simulate each ball. We are generating random number from 0 to 7 both inclusive. 7 means out and 0 to 6 are score on each ball.

First I will show you a code totally written in main method and I have printed it in 1 page using 2 columns; whereas same code is shown in 1 page single column where lot of duplication is handled by an extra function **playinning**. In second code we call method which returns total score of a team. We store it in variable and call same method again and store total score in another variable at the end we compare both results and display result. (Hope you will enjoy both codes)

```

int tScore1=0, score, wkts=0, overs,
balls=0,tScore2=0;
boolean inningInProgress=true;
?ln("First Team Innings:\n");
for (overs=1;inningInProgress;overs++){
    ?("Over No "+overs+":\t");
    balls=1;
    while (balls<=6 && inningInProgress){
        score=(int)(Math.random()*8);
        if (score==7){
            wkts++;
            ?("Out ");
        }
        else{
            tScore1=tScore1+score;
            ?(score+" ");
        }
        if (wkts==6 || (overs==5 && balls==6))
            inningInProgress=false;
        balls++;
    }
    ?ln();
}
?ln("Total Score:"+tScore1);
?ln("Wickets:"+wkts);
?("Overs:");
if (balls<=6)
    ?ln((overs-2)+"."++(balls-1));
else
    ?ln(overs-1);
if (tScore1>tScore2)
    ?ln("Team 1 has won the match");
else if (tScore1<tScore2)
    ?ln("Team 2 has won the match");
else
    ?ln("Match Draw");
?ln("\nSecond Team Innings:\n");
wkts=0;
inningInProgress=true;
for (overs=1;inningInProgress;overs++){
    ?("Over No "+overs+":\t");
    balls=1;
    while (balls<=6 && inningInProgress){
        score=(int)(Math.random()*8);
        if (score==7){
            wkts++;
            ?("Out ");
        }
        else{
            tScore2=tScore2+score;
            ?(score+" ");
        }
        if (wkts==6 || (overs==5 && balls==6))
            inningInProgress=false;
        balls++;
    }
    ?ln();
}
?ln("Total Score:"+tScore2);
?ln("Wickets:"+wkts);
?("Overs:");
if (balls<=6)
    ?ln((overs-2)+"."++(balls-1));
else
    ?ln(overs-1);
if (tScore1>tScore2)
    ?ln("Team 1 has won the match");
else if (tScore1<tScore2)
    ?ln("Team 2 has won the match");
else
    ?ln("Match Draw");

```

```
class CricketDemoUsingFunction{
    public static int playInning(){
        int tScore=0, score, wkts=0, overs, balls=0;
        boolean inningInProgress=true;
        for (overs=1;inningInProgress;overs++){
            System.out.print("Over No "+overs+":\t");
            for (balls=1;balls<=6 && inningInProgress;balls++){
                score=(int)(Math.random()*8);
                if (score==7){
                    wkts++;
                    System.out.print("Out ");
                }
                else{
                    tScore=tScore+score;
                    System.out.print(score+" ");
                }
                if (wkts==6 || (overs==5 && balls==6))
                    inningInProgress=false;
            }
            System.out.println();
        }
        System.out.println("Total Score:"+tScore);
        System.out.println("Wickets:"+wkts);
        System.out.print("Overs:");
        if (balls<=6)
            System.out.println((overs-2)+". "+(balls-1));
        else
            System.out.println(overs-1);
        return tScore;
    }
    public static void main(String []args){
        int tScore1, tScore2;
        System.out.println("First Team Innings:\n");
        tScore1=playInning();
        System.out.println("\nSecond Team Innings:\n");
        tScore2=playInning();
        if (tScore1>tScore2)
            System.out.println("Team 1 has won the match");
        else if (tScore1<tScore2)
            System.out.println("Team 2 has won the match");
        else
            System.out.println("Match Draw");
    }
}
```

For further practice see lab 8 & homework 8 coming soon on website InshaAllah.